Singh and Kaur *J Big Data* (2016) 3:19
DOI 10.1186/s40537-016-0051-6

Journal of Big Data

RESEARCH Open Access

Analyzing performance of Apache Tez and MapReduce with hadoop multinode cluster on Amazon cloud

Rupinder Singh* and Puneet Jai Kaur

*Correspondence:
rupinderkaoni@gmail.com
Department of I.T,
U.I.E.T, Panjab University,
Chandigarh, India

Abstract

Big Data is the term used for larger data sets that are very complex and not easily processed by the traditional devices. Today is the need of the new technology for processing these large data sets. Apache Hadoop is the good option and it has many components that worked together to make the hadoop ecosystem robust and efficient. Apache Pig is the core component of hadoop ecosystem and it accepts the tasks in the form of scripts. To run these scripts Apache Pig may use MapReduce or Apache Tez framework. In our previous paper we analyze how these two frameworks different from each other on the basis of some parameters chosen. We compare both the frameworks in theoretical and empirical way on the single node cluster. Here, in this paper we try to perform the analysis on multinode cluster which is installed at Amazon cloud.

Keywords: Big Data, Hadoop, HDFS, MapReduce, Apache Tez, Apache Pig, Apache Hive

Background

The age of Big Data has begun. Data on servers increased very rapidly and current technologies unable to retrieve some useful information from already stored data [1]. These complex data sets require new technologies so that some useful information is retrieved in timely manner. Many companies invest millions on research to overcome challenges related to Big Data. Apache Hadoop is among the technologies to handle Big Data and it is an open source project maintained by many people around the world [2]. Apache Hadoop foundation has developed many components with different versions. Hortonworks Data Platform is an organization which provides single platform for all the hadoop components [3]. HDP provides us options to install hadoop on different platforms like Microsoft Azure, Amazon cloud, Local site or on own network. Apache Pig is among one of the core components of the hadoop ecosystem. It accepts jobs submitted in the form of scripts. Pig script is saved like notepad file and it is processed line by line using MapReduce or Apache Tez framework. User may choose any framework to run particular pig script. In our previous paper we compare both the frameworks in both theoretical and empirical way on the basis of some parameters. We perform our experiment on the single node cluster and also put more effort on theoretical parameters. Here in this paper we put emphasis on both theoretical empirical parameters and try to analyze that

how these two frameworks react when particular job is submitted to multinode cluster installed on amazon cloud.

Firstly we take closer look at the hadoop ecosystem and some of its components. Then we try to explain parameters used for analysis of both the frameworks. After all the theoretical explanation we try to put some light on Dataset used and experimental setup required for running of Apache Pig Script. We run our script on multinode cluster installed on AWS cloud. Then all the results were shown in the form of graphs and tables. At last we conclude our paper by giving some idea about work yet to be done.

Theoretical analysis

Hadoop ecosystem has many components. Figure 1 shows all the hadoop components for different functions like data access, data management, security, governance and integration. All the hadoop components are very important and have their dedicated roles for solving different type of problems. Here due to space restriction we briefly explain Apache Pig and two frameworks (MapReduce and Apache Tez) required for execution of Pig Scripts.

Apache Pig

Apache Pig is a tool that is used to pre structure the data before hive uses it [4]. It is used for analyzing and transforming datasets. Apache Pig uses procedural and scripting language. Pig job is a series of operations processed in Pipelines and automatically converted into MapReduce Jobs. Pig uses ETL (extract transform model) while extracting data from different sources [5]. Then pig transforms it and stores into HDFS. Pig scripts run on both MapReduce and Apache Tez frameworks. User has three choices to submit pig jobs by grunt shell, UI or java server class.

MapReduce

A few years back we require the single machine for the processing of the larger datasets. Processing data on bigger machines is called scaling up. But this scaling has many bottle necks due to financial and technical issues. To solve this problem the concept of cluster of machines is introduced and this is known as scaling out. To make the concept of distributed processing feasible we have to write new programs. MapReduce is a framework

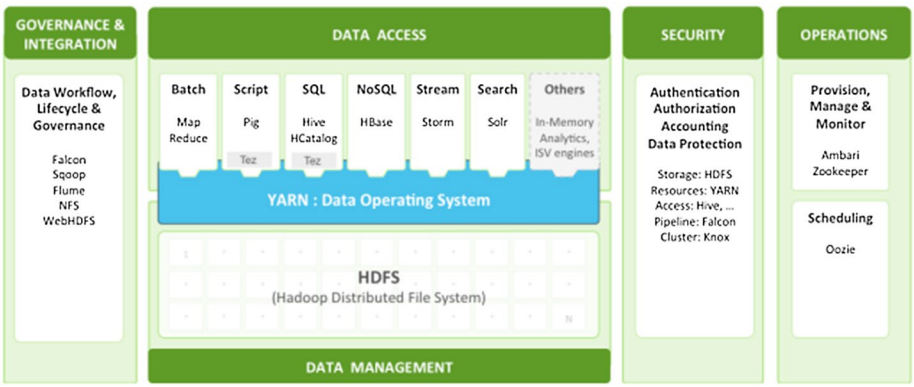


Fig. 1 An illustration of Hadoop ecosystem

which helps in writing programs for processing of data in parallel across thousands of machines [6]. MapReduce is divided into two tasks Map and Reduce. Map phase is followed by the Reduce phase. Reduce phase is always not necessary. MapReduce programs are written in different programming and scripting languages.

Apache Tez

Distributed processing is the base of hadoop. Hive and Pig relies on MapReduce framework for distributed processing. But MapReduce is Batch Oriented. So it is not suitable for interactive queries. So Apache Tez is alternative for interactive query processing. It is available in 2.x versions of Hadoop [3]. Tez is prominent over map reduce by using hadoop containers efficiently, multiple reduce phases without map phases and effective use of HDFS.

Parameters chosen

In our last paper we compare both the frameworks in detail by choosing many parameters and explaining them theoretically [7]. Here we also pick some parameters like execution time and no of containers required by Apache Pig script during its execution. Table 1 shows the comparison of both the frameworks on the basis of parameters chosen.

Experimental evaluation

Apache Tez and MapReduce are two frameworks used by Apache Pig in analysis of particular Dataset [9]. These two frameworks have their own merits and demerits.

Firstly, we discuss about the dataset used in our experiment. Then we explain the experimental setup used for processing of our dataset. At last we discuss the results of analysis of data after running Apache Pig script on both the frameworks.

Dataset used For our analysis we have chosen geographical dataset named GEOLOCATION [3]. In our dataset we have two tables in the form of relational schema having some no attributes. These two tables copied into the hadoop distributed file system and then used by the pig script [10–12]. Readers may ask for more details but due to space constraints we are unable to explain the dataset. So we recommend readers to consult the original reference or mail at authors email id for original dataset. As for study purpose dataset chosen is small but in future we try to perform our analysis on larger dataset. Here due to space constraints we are unable to explain the dataset. So we recommend readers to consult the original reference. Table 2 shows that in our dataset we have 2 relational tables stored in the Apache Hive database.

Experimental setup

Amazon Elastic Compute Cloud (EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. It also provides virtual computing environment also known as instances and different preconfigured templates also known as Amazon machine images (AMI's) for our instances [11]. These AMI's have already installed operating system and other required software. For our cluster we have chosen AMI having Red Hat 6 Linux installed on it and six instances/nodes with different configurations. Ambari server is installed on T1 type instance having one virtual CPU and 500 MB of

Table 1 Difference between MapReduce and Apache Tez on the basis of different parameters

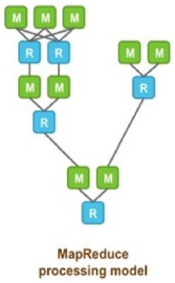
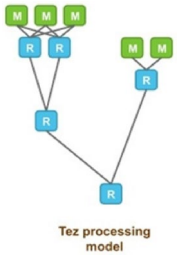
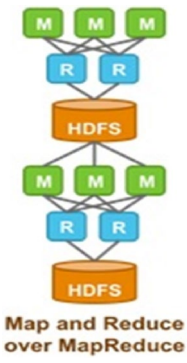
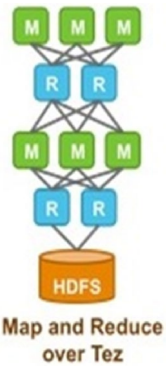
Parameters	MapReduce	Apache Tez
Types of queries	MapReduce supports batch oriented queries [7]	Apache Tez supports interactive queries
Usability	MapReduce is the backbone of hadoop ecosystem and Apache Pig relies on this framework	Apache Tez also works for Apache Pig but it is very useful in interactive scenarios
Processing model	MapReduce always requires a map phase before the reduce phase  MapReduce processing model	A single Map phase and we may have multiple reduce phases  Tez processing model
Hadoop version	MapReduce is backbone of hadoop available in all hadoop versions	Apache Tez is available in Apache Hadoop 2.0 and above
Response time	Slower due to the access of HDFS after every Map and Reduce phase	High due to lesser job splitting and HDFS access
Temporary data storage	Stores temporary data into HDFS after every map and reduce phase [8]  Map and Reduce over MapReduce	Apache Tez doesn't write data into HDFS, so it is more efficient  Map and Reduce over Tez
Usage of hadoop containers	MapReduce divide the task into more jobs. So more containers required for more jobs	Apache Tez reduces this inefficiency by dividing the task into lesser no of jobs and also by using existing containers

Table 2 Datasets used in experiments

Name	No of records	No of attributes
Geolocation	8013	10
Drivermilage	101	2

RAM. Remaining five M3 type nodes are divided into two masters and three slaves. M3 type instances have high frequency Intel Xeon E5-2670 v2 (Ivy Bridge) Processors and SSD based instance storage for faster I/O performance [13]. In our cluster, one different

node for Ambari server is chosen as we don't want that resources of other nodes are wasted in managing the cluster. Figure 2 shows that two master nodes are chosen so that workload is distributed between them and three slave nodes because hadoop maintains the three parallel copies of data. In case of failure of data node, hadoop at the same time start using another copy of data. Putty SSH and telnet client for windows is used to login into shell of operating system of different nodes. T1 type instance configuration is fixed but varying configurations of M3 instances are used for our results. Different configurations of M3 instance are shown in Fig. 3.

Experimental results and metrics

Our dataset is stored in Apache Hive in the form of relation tables [14, 15]. Our Pig Script shown in Fig. 4 is run on both the frameworks and stores the results into the 'riskfactor' table. Riskfactor table is created before running the script. Pig script may be written in different ways and the way of writing the script does not affects our experiment because script is common for both the frameworks. Pig script is executed line by line and temporary results were stored into the variables. This script gives us the results that how risky a driver is. Our main task is to analyze both the frameworks. So we perform analysis depending upon the empirical parameters.

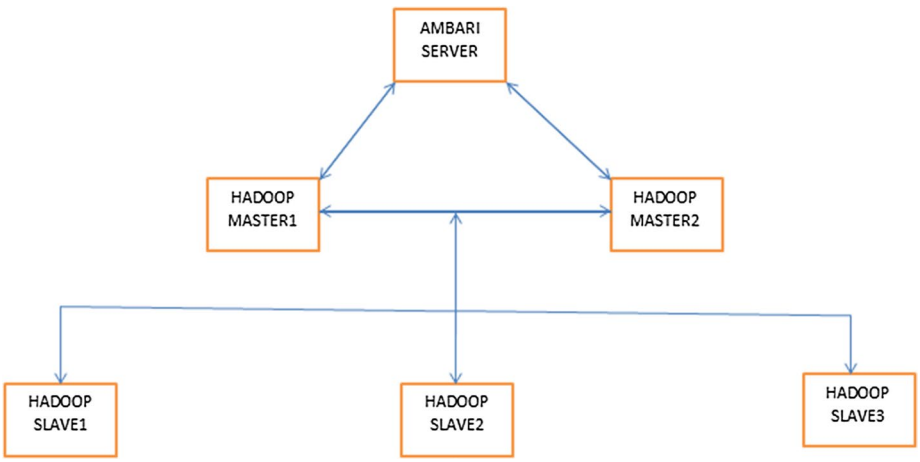


Fig. 2 Boxes show the nodes in our cluster and arrows represent their way of interaction

Model	vCPU	Mem (GiB)	SSD Storage (GB)
m3.medium	1	3.75	1 x 4
m3.large	2	7.5	1 x 32
m3.xlarge	4	15	2 x 40
m3.2xlarge	8	30	2 x 80

Fig. 3 Different type of instances used in our experiment

Apache Script

```
1.  a = LOAD 'geolocation' using
2.  org.apache.hive.hcatalog.pig.HCatLoader();
3.  b = filter a by event != 'normal';
4.  c = foreach b generate driverid, event, (int) '1' as
    occurance;
5.  d = group c by driverid;
6.  e = foreach d generate group as driverid,
    SUM(c.occurance) as t_occ;
7.  g = LOAD 'drivermileage' using
    org.apache.hive.hcatalog.pig.HCatLoader();
8.  h = join e by driverid, g by driverid;
9.  final_data = foreach h generate $0 as driverid, $1
    as events, $3 as totmiles, (float) $3/$1 as
    riskfactor;
10. store final_data into 'riskfactor' using
    org.apache.hive.hcatalog.pig.HCatStorer();
```

Fig. 4 Apache Pig Script used in our experiment

Effect on runtime with increase in configuration of cluster nodes

In our previous paper we perform our experiment on single node cluster with fixed configuration [6]. But here we have multi node cluster with varying configuration. So firstly we run apache pig script ten–ten times on both the frameworks at 3.75 GB, 1 CORE Machines of m3.medium type. Figure 5 shows the results of execution of script on both the frameworks. Vertical axis depicts the time in milliseconds and horizontal axis shows the no of runs of script. Fig clearly shows that Apache Tez has lesser execution time than MapReduce Framework.

We extended our experiment by executing the same script on both the frameworks installed on machines having more no of cores and memory. We choose large, xlarge and 2xlarge machines of m3 type. Figures 6, 7, 8 shows the results for different configurations. It is clearly depicted from all the graphs that Apache Tez has better performance in every case. As the resources available to both the frameworks increased, they move towards the stability and takes same amount of time in every run.

In Fig. 8 we have almost straight lines for both the frameworks and average execution time of script is calculated using the formula

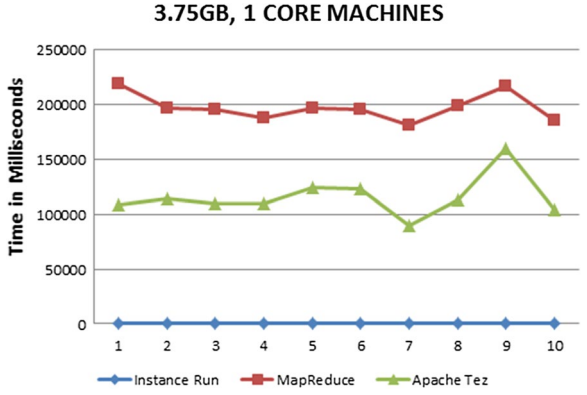


Fig. 5 Results of execution on 3.75GB, 1 core machines

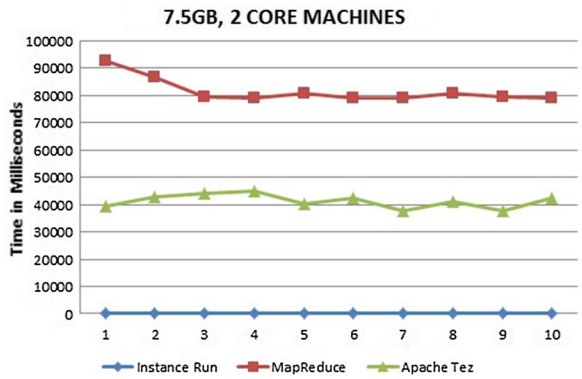


Fig. 6 Results of execution on 7.5GB, 2 core machines

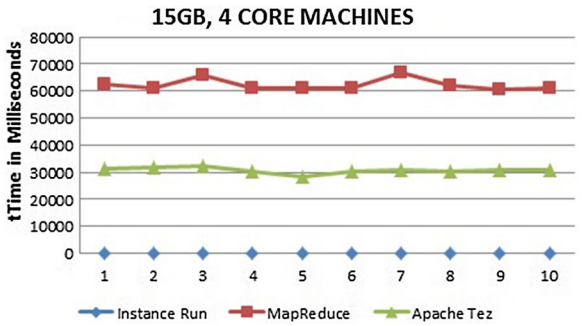


Fig. 7 Results of execution on 15GB, 4 core machines

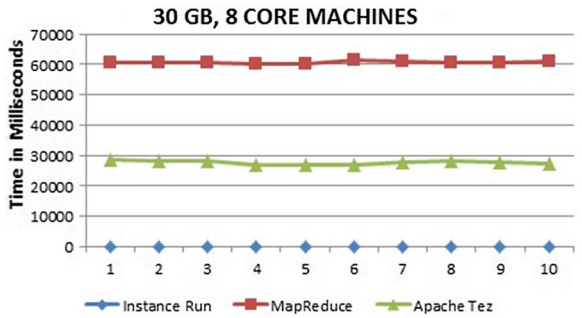


Fig. 8 Results of execution on 30GB, 8 core machines

$$A = \frac{1}{n} * \sum_{i=1}^{n} x_i$$

In our case we have n = 10 and by picking the different values of x from Fig. 8, we got average time of 27,710.7 ms for Apache Tez and 60,713.7 for MapReduce, This shows that MapReduce takes almost double time then Apache Tez.

After execution of script on machines of different configurations we calculated the average time of execution. Figure 9 shows that the execution time decreases as the no of available resources increased and when we move from 15 GB, 4 Core to 30 GB, 8 Core configurations there is slight decrease in slope. This shows that no of resources required

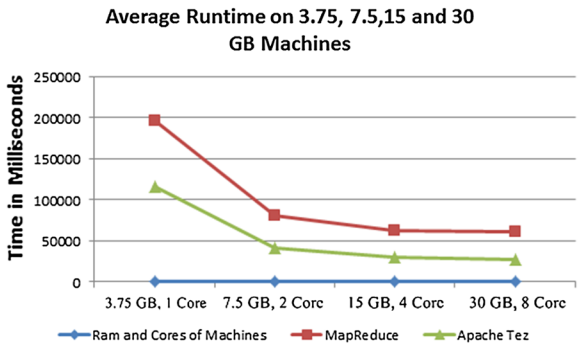


Fig. 9 Average time for execution of script on different machines

for execution of script attains the peak value. No further decrease in shown even if we upgrades to higher configuration.

No of jobs

The Apache Pig script is submitted to both the frameworks. Apache Tez completed the whole script as a one job while MapReduce divides it into two jobs. Due to two jobs MapReduce takes more time than Apache Tez. Hadoop memory containers firstly allocated for one job and second job got chance only after completion of the first job. Flowcharts for Apache Tez and MapReduce in Fig. 10 show the process of allocation and deallocation.

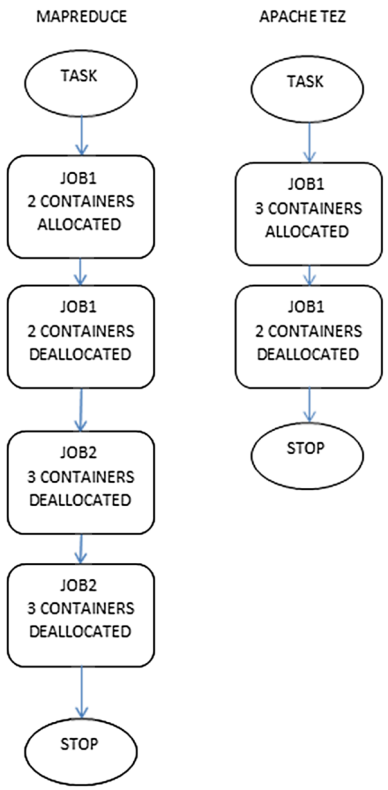


Fig. 10 Flowchart of hadoop containers allocation

Table 3 Showing difference on the basis of some parameters

Parameter chosen	MapReduce	Apache Tez
No of jobs	2 jobs	1 job
No of containers	1st job = 2 containers 2nd job = 3 containers	1 job = 3 containers

No of containers required

Hadoop containers are the pool of resources allocated to particular job during its execution. Apache Tez and MapReduce required different no of containers during execution of same script. In previous section we already explained that MapReduce requires more no of containers then Apache Tez. Table 3 shows the no of jobs and hadoop containers required by both the frameworks.

Conclusion and future work

This paper briefly explains both the frameworks used for execution of Pig Scripts. We try to perform both theoretical and empirical analysis on the basis of some parameters. With the help of chosen parameters we are able to understand that how these frameworks differ from each other. Results show that Apache Tez is a better choice for execution of Apache Pig scripts as MapReduce requires more resources in the form of time and storage. But MapReduce is also the backbone of hadoop ecosystem and can be used efficiently in various scenarios.

In future we try to go into more detail of Apache Tez framework and try to explore new things so that it becomes more efficient. Lots of work is yet to be done on this framework.

Authors' contributions

RS performed the primary literature review, data collection, experiments, and also drafted the manuscript. PJK worked with RS and helps in analyzing the frameworks. Both authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Received: 18 June 2016 Accepted: 29 August 2016
Published online: 18 October 2016

References

1. Big Data, http://www.sen.wikipedia.org/wiki/Big_data.
2. Hadoop, http://www.apache.org.
3. Hadoop, http://www.hortonworks.com.
4. Ouaknine K, Carey M, Kirkpatrick S. The Pig mix benchmark on Pig, MapReduce, and HPCC systems. In: 2015 IEEE international congress on Big Data (BigData Congress); 2015. p. 643–8.
5. Bansal SK. Towards a semantic extract-transform-load (ETL) framework for Big Data integration. In: 2014 IEEE international congress on Big Data (BigData Congress); 2014. p. 522–9.
6. Maitrey S, Jha CK. Handling Big Data efficiently by using MapReduce technique. In: IEEE international conference on computational intelligence & communication technology (CICT); 2015. p. 703–8.
7. Singh R, Kaur PJ. Theoretical and empirical analysis of usage of MapReduce and Apache Tez in Big Data. In: Proceedings of first international conference on information and communication technology for intelligent systems: volume 2, smart innovation, systems and technologies 51, doi: 10.1007/978-3-319-30927-9_52.
8. Ravindra P. Towards optimization of RDF analytical queries on MapReduce. In: IEEE 30th international conference on data engineering workshops (ICDEW); 2014. p. 335–9.
9. Fuad A, Erwin A, Ipung HP. Processing performance on Apache Pig, Apache Hive and MySQL cluster. In: 2014 international conference on information, communication technology and system (ICTS); 2014. p. 297–302.

10. Azzedin F. Towards a scalable HDFS architecture. In: 2013 international conference on collaboration technologies and systems (CTS); 2013. p. 155–61.
11. Gates AF, Dai J, Nair T. Apache Pig's optimizer. IEEE Data Eng Bull. 2013;36(1):34.
12. Gates AF, Natkovich O, Chopra S, Kamath P, Narayanamurthy SM, Olston C, Reed B, Srinivasan S, Srivastava U. Building a high-level dataflow system on top of Map-Reduce: the Pig experience. Proc VLDB Endow. 2009;2(2):1414–25.
13. Cluster, http://www.amazon.com.
14. Thusoo A, Sarma JS, Jain N, Shao Z, Chakka P, Zhang N, Antony S, Liu H, Murthy R. Hive-a petabyte scale data warehouse using hadoop. In: 2010 IEEE 26th international conference on data engineering (ICDE). New York: IEEE; 2010. p. 996–1005.
15. Thusoo A, Sarma JS, Jain N, Shao Z, Chakka P, Anthony S, Liu H, Wyckoff P, Murthy R. Hive: a warehousing solution over a map-reduce framework. Proc VLDB Endow. 2009;2(2):1626–9.